

**Link: https://editor.plantuml.com**

**Link aufrufen, Code rauskopieren, png image anklicken, speichern/download**

@startuml

' --- Gesamteinstellungen ---

title Klassendiagramm

!theme vibrant

skinparam linetype ortho

skinparam classAttributeIconSize 1

skinparam packageStyle rectangle

hide empty members

' --- Farbpaletten ---

skinparam title {

BackgroundColor #F5F5F5

BorderColor #5D6D7E

FontColor #34495E

}

skinparam package {

BackgroundColor #EBF5FB

BorderColor #85C1E9

}

skinparam package<<Frame>> {

BackgroundColor #EAECEE

BorderColor #AEB6BF

}

skinparam package<<Strategy>> {

BackgroundColor #E8F8F5

BorderColor #76D7C4

}

skinparam package<<Cloud>> {

BackgroundColor #FEF9E7

BorderColor #F8C471

}

skinparam class {

BackgroundColor #FFFFFF

BorderColor #5D6D7E

ArrowColor #5D6D7E

}

skinparam interface {

BackgroundColor #D5F5E3

BorderColor #58D68D

}

skinparam enum {

BackgroundColor #FCF3CF

BorderColor #F1C40F

}

skinparam abstract {

BackgroundColor #FDEDEC

BorderColor #F1948A

}

skinparam class<<Application>> {

BackgroundColor #D6EAF8

BorderColor #5499C7

}

skinparam class<<Script>> {

BackgroundColor #D5DBDB

BorderColor #839192

}

' =======================================

' ===== MESA FRAMEWORK (Referenzen) =====

' =======================================

package "Mesa Framework" <<Frame>> {

abstract class Model

abstract class CellAgent

class FixedAgent

class OrthogonalMooreGrid

class DataCollector

class ABMSimulator

}

' ====================================

' ===== SIMULATION CORE (Logik) =====

' ====================================

package "Simulation Core" {

' --- Hauptmodell ---

class Exploration {

+ seed: int

+ simulator: ABMSimulator

+ height: int

+ width: int

+ obstacle\_density: float

+ no\_robots: int

+ initial\_no\_robots: int

+ grid: OrthogonalMooreGrid

+ pubSubBroker: PubSubBroker

+ robot\_type: class

+ no\_unexplorable: int

+ datacollector: DataCollector

+ running: bool

+ \_\_init\_\_(...)

+ step()

- \_place\_obstacles()

}

' --- Komponenten ---

package "Components" {

class PubSubBroker {

- subscribers: defaultdict

+ subscribe(topic, callback, agent\_id)

+ unsubscribe(topic, callback)

+ publish(topic, data, sender\_id)

}

class LocalMemory {

+ grid\_info: dict

+ frontier\_info: dict

+ environment\_grid\_width: int

+ environment\_grid\_height: int

+ get\_known\_neighbor\_positions(pos)

+ get\_all\_neighbor\_positions(pos)

}

' --- Datenstrukturen ---

package "Data Structures" {

class AgentInfo <<dataclass>>

class CellInfo <<dataclass>>

class FrontierInfo <<dataclass>> {

+ status: FrontierStatus

+ agent\_id: Optional<int>

}

enum FrontierStatus {

IN\_WORK

OPEN

}

}

}

' --- Agenten ---

package "Agents" {

package "Robots" {

abstract class ExplorerRobot {

+ {abstract} step()

+ {static} normalize\_round45\_angle(angle)

+ scan\_environment()

- {static} \_bresenham\_line(start, end)

- {static} \_angle\_generator(angle, resolution)

}

class FBERobot {

+ pubSubBroker: PubSubBroker

+ pathfinder: Pathfinder

+ goal\_finder: MovementGoalFinder

+ goal\_selector: MovementGoalSelector

+ goal: tuple

+ path: list

+ step()

- \_new\_gird\_info\_callback(data)

- \_new\_frontier\_info\_callback(data)

}

class RandomWalkRobot {

+ step()

}

}

package "Grid Agents" {

class Obstacle

class Ground

}

}

' --- Algorithmen ---

package "Algorithms" <<Strategy>> {

package "Pathfinding" {

interface Pathfinder {

{abstract} find\_path(goal\_pos)

}

class AStar {

- agent: ExplorerRobot

- heuristic: Callable

+ find\_path(goal\_pos)

- \_reconstruct\_path(end\_node)

}

class Node {

+ pos: tuple

+ real\_costs\_from\_start: float

+ estimated\_costs\_till\_goal: float

+ parent: Node

+ cost: float

}

class PathfinderFactory {

+ {static} give\_pathfinder(agent, name)

}

enum PathfinderEnum {

ASTAR

}

}

package "Movement Goal Finding" {

interface MovementGoalFinder {

{abstract} find\_goals()

}

class OriginalFBE {

- agent: ExplorerRobot

+ find\_goals()

- \_is\_frontier(pos, cell\_info)

}

enum MovementGoalFinderEnum {

ORIGINAL\_FBE

}

class MovementGoalFinderFactory {

+ {static} give\_movement\_goal\_finder(agent, name)

}

}

package "Movement Goal Selection" {

interface MovementGoalSelector {

{abstract} select\_goal(goals)

}

class NearestBiggestFrontier {

- agent: ExplorerRobot

- factor\_distance: float

- factor\_size: float

+ select\_goal(goals)

- \_connected\_component\_clustering(positions)

}

enum MovementGoalSelectorEnum {

NEAREST\_BIGGEST\_FRONTIER

}

class MovementGoalSelectorFactory {

+ {static} give\_movement\_goal\_selector(agent, name, ...)

}

}

}

}

' ==================================================

' ===== APPLICATION & RUNNERS (Nutzung) =====

' ==================================================

package "Application & Runners" <<Cloud>> {

class "SolaraViz App (app.py)" <<Application>>

class "Batch Runner (batch.py)" <<Script>>

}

' ===========================

' ===== BEZIEHUNGEN =====

' ===========================

' --- Vererbung & Implementierung ---

Model <|-- Exploration

CellAgent <|-- ExplorerRobot

FixedAgent <|-- Obstacle

FixedAgent <|-- Ground

ExplorerRobot <|-- FBERobot

ExplorerRobot <|-- RandomWalkRobot

Pathfinder <|.. AStar : <<implements>>

MovementGoalFinder <|.. OriginalFBE : <<implements>>

MovementGoalSelector <|.. NearestBiggestFrontier : <<implements>>

' --- Abhängigkeiten & Assoziationen ---

Exploration "1" o-- "1" OrthogonalMooreGrid

Exploration "1" \*-- "1" PubSubBroker

Exploration ..> "explorability\_analysis"

' Robot -> Components

FBERobot "1" -- "1" PubSubBroker : uses

ExplorerRobot "1" \*-- "1" LocalMemory

' Robot -> Algorithm Interfaces (Strategy Pattern)

FBERobot "1" o-- "1" Pathfinder

FBERobot "1" o-- "1" MovementGoalFinder

FBERobot "1" o-- "1" MovementGoalSelector

' Robot -> Factories

FBERobot ..> PathfinderFactory

FBERobot ..> MovementGoalFinderFactory

FBERobot ..> MovementGoalSelectorFactory

' Factories -> Concrete Algorithms & Enums

PathfinderFactory ..> PathfinderEnum

PathfinderFactory ..> AStar : <<creates>>

MovementGoalFinderFactory ..> MovementGoalFinderEnum

MovementGoalFinderFactory ..> OriginalFBE : <<creates>>

MovementGoalSelectorFactory ..> MovementGoalSelectorEnum

MovementGoalSelectorFactory ..> NearestBiggestFrontier : <<creates>>

' Concrete Algorithms -> Other Classes

AStar ..> ExplorerRobot

AStar ..> Node : <<creates>>

OriginalFBE ..> ExplorerRobot

NearestBiggestFrontier ..> ExplorerRobot

' Application -> Core

"SolaraViz App (app.py)" ..> Exploration

"Batch Runner (batch.py)" ..> Exploration

@enduml